ANSYS / CFX

ANSYS Workbench

ANSYS workbench can be started with the following command.

```
runwb2
```

ANSYS Mechanical APDL

ANSYS Mechanical APDL can be startet with the following command (replace the number in the binary name ansys231 with the appropriate version you use; this example is done after a `module load ANSYS/2023.1`):

```
ansys231
```

Likewise an interactive session in graphics mode can be started with the following command.

```
ansys231 -g
```

Starting Ansys on one node (shared memory) from a job script:

```
..
#SBATCH --cpus-per-task=12
..
export ANSWAIT=1
ansys221 -b -np $SLURM_CPUS_PER_TASK -i test.dat -o test.out
```

To use cfx5solve:

```
cfx5solve -batch -def mytest.def  -par-dist $nodes -start-method "Open MPI
Local Parallel"
```

Starting Ansys on multiple nodes (distributed memory) from a job script; Attention: fill up complete nodes before you start using multiple nodes. Communication between nodes takes much longer than intra-node, so try to stay on one node as long as you can.

```
..
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=32
..
export ANSWAIT=1
ansys221 -b -dis -np $SLURM_NTASKS -machines $(expand-slurm-nodelist -m) -i
test.dat -o test.out
```

Submit an interactive cfx5solve job to multiple nodes (distributed parallel); again, try to stay on one node as long as you fit onto one machine, only use multiple nodes when ressources needed are too

large:

```
salloc --nodes=2 --ntasks-per-node=32 --mem-per-cpu=3G --time=6:0:0
```

As soon as the nodes have been allocated:

```
module  load ANSYS/2021.2
nodes=$(expand-slurm-nodelist --cfx)
cfx5solve -batch -def mytest.def  -par-dist $nodes -start-method "Open MPI
Distributed Parallel"
```

ANSYS Tips

Problems

In case you get a message “Unexpected error: The following required addins could not be loaded: Ans.SceneGraphChart.SceneGraphAddin. The software will exit.”, try `module load foss/2021b Mesa/.21.1.7` first. The problem seems to be specific to Ansys/2025.1, which seems to have incompatibilities with the default OpenGL environment. For X2GO, a workaround (loading module “StdEnv”, which is only available on the login nodes and which basically just loads the hidden module Mesa/.21.1.7 that seems to work with Ansys/2025.1) is already in place. In the web portal (“OOD”, OpenOnDemand, the thing you log into by pointing your browser to https://login.cluster.uni-hannover.de), one needs to work around manually loading Mesa/.21.1.7, a prerequisite of which is e.g. the above mentioned foss/2021b module (or the specific GCC version loaded by this version, to be more precise). Check for the module using `module –show-hidden spider Mesa` in case you are interested.

Memory usage

Many errors are due to jobs not being configured properly, in particular requesting not enough memory. The system checks the memory request you made at several instances and from time to time, and in case you overstep what you requested, your job will/may get killed (it may, at times, make it through, however, giving rise to questions like “but it ran without problems up to now” - which is not true, but you just did not yet see the problems and the system did not yet kill your jobs).

So please try to adapt your job to your requirements. To find out how the nodes are configured, see our table of computing hardware. Try to use about the same fraction of memory as the number of cpu cores on a node to facilitate matching those hardware components - our computers internally usually consist of several so-called NUMA-nodes, which means that each cpu-socket also has RAM that is directly attached to it, and this usually is the fraction of memory that corresponds to the fraction of cpu cores the socket contains. Use slightly less than the maximum amount of memory in that fraction to leave room for the operating system (Linux) and some buffers so your job doesn't have to be squeezed. 4 GB should be enough, 8 GB may see some improvements, and your exact mileage may vary.

Node usage

If you can, you should try to stay on one node instead of requesting fractions of several nodes. Inter-node communication usually takes much longer than intra-node communication, so you may benefit from filling up nodes first and only expanding to other nodes when the job gets too big to fit on one node. We see, of course, that one may fit in faster by requesting only fractions of nodes, but that may not deliver the best overall performance. And if you occupy only parts of nodes, you'll also make it more difficult for others to get full nodes.

Enos partition equipped with OmniPath instead of Infiniband

You might come across an error like this when running ANSYS on enos nodes.

```
ansysdis201: Rank 0:8: MPI_Init_thread: multiple pkey found in partition key
table, please choose one via MPI_IB_PKEY
```

Enos nodes of the cluster system do not have an InfiniBand interconnect but use Omni-Path instead. If you would like to run ANSYS on enos nodes, chose the correct partition key (pkey) by adding the following line to your job script before calling the ANSYS application.

```
[[ $HOSTNAME =~ ^enos-.* ]] && export MPI_IB_PKEY=0x8001
```

However, sometimes ANSYS, or the underlying mpi-implementation used, does not seem to honour the exported variable causing the error to persist. Unfortunately the ANSYS documentation on their mpi implementations is scarce. In this case please contact ANSYS support or exclude enos nodes from your job in order to circumvent the error. You can write a PARTITION line in your resource specification with all partitions you would like to use except enos. There is no option to exclude enos or any other partition.

Debugging

To see what ansys does when it runs, set `export ANS_SEE_RUN_COMMAND=1`

Some Sample Scripts

Case 1: SMP (Shared Memory Parallel). Best for single-node jobs where all cores share the same memory space

```
#!/bin/bash -l
#SBATCH --job-name=ansys_smp
#SBATCH --nodes=1              ## SMP must stay on 1 node
#SBATCH --cpus-per-task=40     ## Number of CPU cores for threading.
                               ## 40 fits well on amo cluster nodes
#SBATCH --mem=64G              ## Adjust based on model size. Set mem=0
                               ## if you reserve the whole node
```

```
#SBATCH --time=12:00:00

# Load ANSYS environment
module load  ANSYS/2024.2

# Run ANSYS mechanical in SMP mode
cd $SLURM_SUBMIT_DIR
ansys261 -b -smp -np $SLURM_CPUS_PER_TASK -i input.dat -o output.out
```

Case 2: Distributed Memory Parallel. For large models that require multiple nodes

```
#!/bin/bash -l
#SBATCH --job-name=ansys_dis
#SBATCH --distribution=cyclic
#SBATCH --nodes=2             ## Can span multiple nodes
#SBATCH --ntasks-per-node=40
#SBATCH --mem-per-cpu=4G
#SBATCH --time=12:00:00

# Load ANSYS environment
module load ANSYS/2024.2

# Run ANSYS mechanical in distributed mode
cd $SLURM_SUBMIT_DIR
ansys242 -b -mpi openmpi -dis -np $SLURM_NTASKS -i input.dat -o output.out
```

To monitor your running processes once the job starts, ssh into the compute node and run: `htop -u $USER`

From:
https://docs.cluster.uni-hannover.de/ - **Cluster Docs**

Permanent link:
https://docs.cluster.uni-hannover.de/doku.php/guide/soft/ansys

Last update: **2026/01/26 19:11**